

Reference architecture design for farm management information systems: a multi-case study approach

J. Tummers¹ · A. Kassahun¹ · B. Tekinerdogan¹

Published online: 1 June 2020 © The Author(s) 2020

Abstract

One of the key elements of precision agriculture is the farm management information system (FMIS) that is responsible for data management, analytics and subsequent decision support. Various FMISs have been developed to support the management of farm businesses. A key artefact in the development of FMISs is the software architecture that defines the gross level structure of the system. The software architecture is important for understanding the system, analysing the design decisions and guiding the further development of the system based on the architecture. To assist in the design of the FMIS architecture, several reference architectures have been provided in the literature. Unfortunately, in practice, it is less trivial to derive the application architecture from these reference architectures. Two underlying reasons for this were identified. First of all, it appears that the proposed reference architectures do not specifically focus on FMIS but have a rather broad scope of the agricultural domain in general. Secondly, the proposed reference architectures do not seem to have followed the proper architecture documentation guidelines as defined in the software architecture community, lack precision, and thus impeding the design of the required application architectures. Presented in this article is a novel reference architecture that is dedicated to the specific FMIS domain, and which is documented using the software architecture documentation guidelines. In addition, the systematic approach for deriving application architectures from the proposed reference architecture is provided. To illustrate the approach, the results of multi-case study research are shown in which the presented reference architecture is used for deriving different FMIS application architectures.

Keywords Farm management information system \cdot Reference architecture \cdot Architecture design \cdot Multi-case study research

Information Technology Group, Wageningen University & Research, Hollandseweg 1, 6706 KN Wageningen, The Netherlands



B. Tekinerdogan bedir.tekinerdogan@wur.nl

Introduction

The continuous developments in information technology (IT) over the last decades have a substantial and increasing impact on various industrial domains. In recent years, IT and the various developments such as the internet of things (IoT), cloud computing and machine learning have also shaped the farming domain in which digitization is increasingly adopted to enhance the effective and efficient realization of the various farm business processes. For example, with the help of IoT, farming practices such as yield monitoring, cultivar selection, pest management, irrigation, etc. can be applied more precisely (Köksal and Tekinerdogan 2019). In this context, farm management information systems (FMISs) are being developed to manage the large amount of information that is involved in these processes, and likewise to keep track of and support the farm activities. The rapid development of FMISs over recent years has been partly driven by progress in precision agriculture (Hewage et al. 2017). FMISs can play a significant role in precision agriculture by providing functions for site-specific purposes (Fountas et al. 2015a) and can reunite the different parts needed for precision agriculture (Nikkilä et al. 2010; Stafford 2000). Each FMIS supports various features such as milk yield management in dairy farming, fertilization management in arable farming and financial management in all types of farming. Examples of FMISs are Agworld, FarmWorks or 365FarmNet (Paraforos et al. 2016). Each FMIS focuses on one or multiple domains of the agricultural sector, for example, livestock or arable farming (Tummers et al. 2019).

One of the critical artefacts in the design of FMISs is the software architecture (Köksal and Tekinerdogan 2019; Tekinerdogan and Köksal 2018). Bass et al. (2003) defined the software architecture of a program or a computing system as:"The structure of the system, which comprise software components, the externally visible properties of those components and the relationships among them." An architecture-based development of FMISs has several benefits including support for communication among the stakeholders, guidance of the design decisions and evaluation of the system. For proper designing and documenting of the software architecture, usually the guidelines as defined in the software architecture domain are followed. With this, software architecture is documented according to the various stakeholder concerns and, for this reason, multiple different so-called software architecture views are used (Clements et al. 2010). In this case, for the agricultural domain, multiple stakeholders have their concerns, which can be mapped using specific architecture views. A particular type of software architecture that is generic and can be used to help design specific software architectures of multiple software systems is the reference architecture (RA). An RA is a generic design that can be used to derive specific application architecture (AAs) based on the identified stakeholder concerns. RAs can help to design AAs more quickly and also with higher quality. For this, it is needed that the RA itself defines the proper scope of the domain and is well-documented.

The objective of this article is to develop a validated RA dedicated to FMISs following well-established architecture design methods. This study builds on the results of an earlier study in which a systematic literature review (SLR) was conducted on the state-of-the-art of FMISs (Tummers et al. 2019). In the earlier study, features that are implemented in current FMISs and obstacles that are related to the development and usage of FMISs were systematically identified. In the literature, several RA designs have been proposed for FMIS. Unfortunately, in practice, it seems less trivial to derive the AA from these RAs. Two basic reasons could be identified for this. First of all, it appears that the proposed RAs do not specifically focus on FMIS but have a rather broad scope of the agricultural domain



in general. Secondly, the proposed RAs do not seem to follow the proper architecture documentation guidelines as defined in the software architecture community, lack precision and thus impede the design of the required AAs.

To fulfil the objective of this article, a new RA is proposed that was developed based on the findings of the earlier study and experiences in designing software architectures. The RA is dedicated to the specific FMIS domain and is documented using the software architecture documentation guidelines. In addition, the systematic approach for deriving AAs from the proposed RA is provided. To illustrate the approach and to validate the RA, a multi-case study protocol was conducted in which different AAs were derived from the proposed RA. Based on these experiences, the experiences and lessons learned for deriving the AA from the proposed RA are reported.

Background

This section presents the summary of the results of the earlier SLR (Tummers et al. 2019) that was conducted, and then describes the background on RA design and documentation.

FMIS

Currently, many different FMISs have been proposed that can be used to support farm management activities (Fountas et al. 2015a; Capterra 2020; Köksal and Tekinerdogan 2019). According to Sørensen et al. (2010), an FMIS is defined as: "a planned system for the collecting, processing, storing and disseminating of data in the form of information needed to carry out the operations functions of the farm." The FMIS is a management information system (MIS) for the agricultural domain. Waston et al. (1991) defined the MIS as: "an organizational method of providing past, present and projected information related to internal operations and external intelligence." Over the years, FMISs have developed from simple recording systems into extensive FMISs (Fountas et al. 2015a). Where the MIS can support decision making by providing timely information about the planning, control and operational functions of an organization (Waston et al. 1991), the FMIS does the same for the agricultural domain. Currently, the primary goal of FMISs is to reduce the production costs, maintain high quality and to comply with agricultural standards (Fountas et al. 2015a).

In Tummers et al. (2019), an SLR was conducted in order to identify the state-of-theart of FMISs. An SLR makes it possible to identify, evaluate and interpret all available research relevant to a particular research question, topic area or phenomenon of interest (Kitchenham et al. 2009). With the help of a review protocol based on the guidelines presented in Kitchenham et al. (2009), this study identified 1048 papers of which 38 primary studies were selected and analysed in further detail, these are presented in A. This SLR documented the commonly used features and the commonly encountered obstacles to the development and adoption of FMISs. A feature was described as a user-visible characteristic of an FMIS, and an obstacle was described as a problem related to the development or the use of FMISs. These 38 studies have all been published since 2008 and passed various selection and quality selections. With the detailed analysis, 81 unique FMIS features, 53 unique obstacles of FMISs and multiple (reference) architectures were identified. The features are presented in Appendix B, the obstacles in Appendix C and the RAs are further discussed in the "Current reference architectures" section. Furthermore, a unique set of 22



stakeholders and their concerns regarding the development of FMISs was identified, which are presented in Table 1. For this table, the stakeholders relevant to the development of an FMIS were considered. Herein the definition of stakeholder given by the project management institute (PMI) (project management institute 2013) was adopted: "an individual, group, or organization, who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project".

To model the common and variant features of the FMIS domain, feature modelling was adopted. In the feature model, a "feature" is defined as "a prominent or distinctive

Table 1 The identified stakeholder and their concerns. Adapted from Tummers et al. (2019)

Role	Concerns
Farmer	Responsible person on the farm and end-user of the system
Government	Has an interest in FMISs for registration purposes, and to obtain farm information
Agricultural expert	Has expert knowledge about the agricultural sector and can be used for requirements for FMISs
Farm employee	Works on the farm and has to work with the FMIS
Research Institute	Multiple kinds of researchers and institutes can be used as knowledge input for FMISs
FMIS developer	Develops the FMIS and its underlying software
Input supplier	Delivers inputs to the farm; these inputs can be registered in an FMIS
Agricultural advisor	Helps the farmer with making decisions based on their knowledge, an FMIS can assist them
Agriculture service provider	Assists the farmer with the provision of services. Can use FMIS for registration purposes
Contractor	Hired by the farmer to perform field tasks. FMISs can improve communication with the farmer
Equipment producer	Makes new machinery for the farmer, an FMIS can provide machinery management
Customer	Companies and other entities greater than the individual consumer. FMISs can provide details about the purchased products
Administrator	Can set up system, and manages the FMIS. Is not necessarily the FMIS developer
Farmers association	Organised group of farmers with common interests. Wants FMIS for the implementation of modern technology
Neighbour	Can be influenced by decisions of FMIS (odour nuisance, noise disturbance, etc.)
Non-governmental	Group of persons with their own (ecological) interest that can be intertwined with the FMIS
Product processor	FMIS can provide information on products coming from the farm
Veterinarian	Can use the FMIS for retrieving animal information and can register veterinarian actions
Accountant	Can use the financial modules of FMISs to verify and assist the farmer with bookkeeping
Equipment dealer	Can provide machinery support and services via the FMIS
Media	Provides communication with the outside world and has an influence on the farm image
Weather service provider	Provides weather information as input for the FMIS



user-visible aspect, quality or characteristic of a software system or system" (Kang et al. 1990). The feature model is a tree-shaped model that shows the common, alternative and optional features of a system (Kang et al. 1990). The feature model for the FMIS is presented in Fig. 1 and shows the features for the FMIS split up into four main features: general MIS, data management, crop farming and animal farming. The general MIS feature contains the features that are not agriculture-specific and are found in MISs over multiple sectors. The data management feature contains features that are related to the management of the data and data-based decision making. The crop farming feature contains the sub-features directly related to the farming of crops, and the animal farming feature contains only the features that are directly related to the management of animals. The features are based on the features from the previous study (available in Appendix B) where features that could be sub-features of another feature are shown under the most general feature. The precise definition of the features can be seen in the previous SLR (Tummers et al. 2019), where the sixteen most occurring features from the literature were described.

Architecture design

Every software system of interest has an architecture that determines its structure. The software architecture is an abstract representation of a system that presents the gross-level structure. The architecture is important for supporting communication among stakeholders, for guiding design decisions and for analysis of the overall system (Tekinerdogan and Uzun 2019). There are two traditions of software architecture design: informal and formal. Informal architecture design does not follow a particular modelling technique and uses simple boxes-and-lines models for representing the architecture. The formal and well-established approach uses multiple views of an architecture description following the ISO/ISEC/IEEE 42010 standard (ISO/IECE 42010 2011).

The conceptual model for the core architecture description from the ISO/ISEC/IEEE standard is presented in Fig. 2. The architecture description has one or multiple views, and each view must have a viewpoint. Each view addresses one or more of the stakeholder concerns. The architecture view describes the architecture of a system in alignment with the conventions and rules in its architectural viewpoint.

According to Clements et al. (2010) and their views and beyond (V&B) approach, each view has a different style. They divide the views into four styles: module, componentand-connector, allocation, and hybrid style. Each style can again be divided into a total of seventeen sub-styles. The module style addresses concerns related to the implementation where the component-and-connector style has views related to the interaction structure. The allocation style has views that describe how software elements are allocated to the environment of the system. A design that can generalise multiple software architectures from a specific domain is an RA. According to the U.S. department of defense, an RA is: "an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions (US Dept. of Defence/ Office of the DoD CIO 2010)." This source of information can be obtained by documenting the relevant views for the architecture, where multiple views together make the RA. The RA should serve as an architecture blueprint for future architects and should provide a common lexicon, taxonomy and (architectural) vision (Muller 2012). Furthermore, the RA should encourage the use of common standards, specifications and patterns for the FMIS (US Dept. of Defence/Office of the DoD CIO 2010).



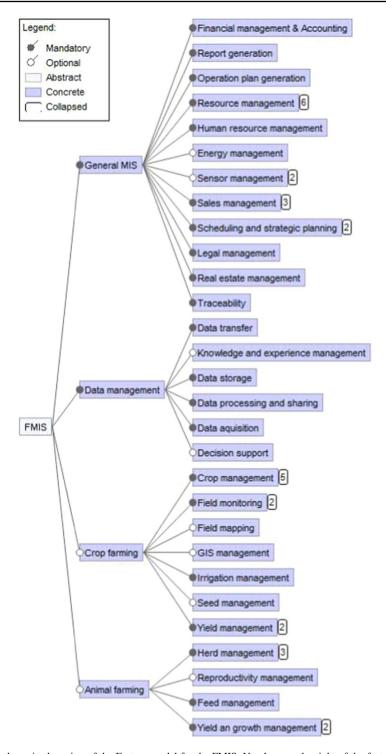


Fig. 1 A downsized version of the Feature model for the FMIS. Numbers on the right of the features show the number of sub-features not shown



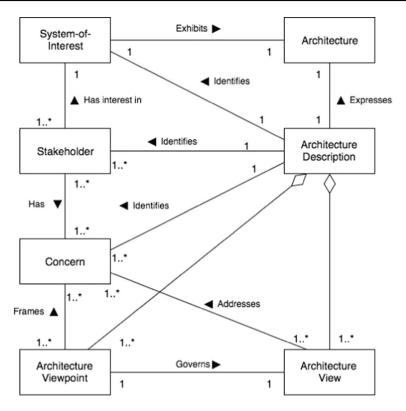


Fig. 2 Part of the content model of the core architecture description. Adapted from the ISO/ISEC/IEEE 42010 standard (ISO/IEC/IEEE 42010, 2011)

RAs are usually developed either through a collaboration of diverse organizations (through a "committee") or by an organization that has multiple and diverse customers. To be useful, the RA should obey the following criteria (Muller 2012): Understandable for all stakeholders, accessible and read/seen by the majority of the organization, addresses the critical domain issues, satisfactory quality, acceptable, up-to-date and maintainable, adds value to the business. According to Kassahun (2017), there are three different scenarios to apply an RA which are presented in Fig. 3. The RA can be used as a reference for the development of an AA. In this study, the AA is defined as the software model of a particular application presented by a combination of architectural views. First, an RA needs to be made, which can be used for the development of the AA. The RA consists of multiple views, which can each be used for the derivation of the same views in the AA. Each selected set of views for the RA will therefore also lead to a corresponding set of views for the AA as can be seen in Fig. 4.



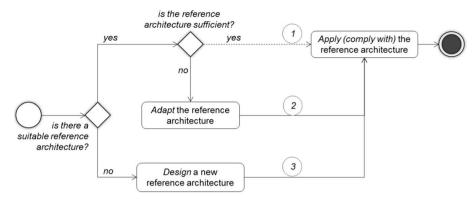


Fig. 3 Three scenarios of applying an RA. Adopted from Kassahun (2017)

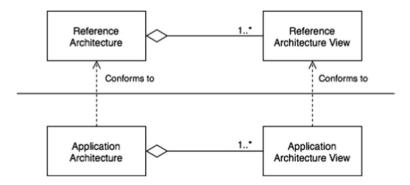


Fig. 4 Methods used for deriving the AA. Each view from the RA will lead to a view in the AA

Research methodology

To address the earlier defined issues on the proper scoping and documentation of RAs, the objective of this study was to develop an RA dedicated to FMIS and documented according to the current architecture documentation guidelines. To fulfil this objective, the following research questions were defined.

- RQ1: What are the existing reference architectures for FMISs?
- RQ2: How to design a feasible reference architecture for FMISs?
- RQ3: How can an application architecture be derived from the reference architecture?
- RQ4: How effective is the designed reference architecture for deriving application architectures?

The first research question focuses on identifying the preliminary work on RAs for FMISs. This research question will be answered using the studies coming from the SLR that focus on architectures. The results of this research question are available in the "Current reference architectures" section. The second research question aims to derive a feasible RA that covers both a broad domain of systems and is designed according to the existing well-established architecture design knowledge. For this, design science research was used



and the guidelines for designing architectures as they are defined in the software and system architecture design community. The results for this research question are presented in the "Reference architecture" section.

The third research question aims to guide developers to derive an AA from the RA; the results of this question are presented in the "Method for deriving an application architecture" section. The fourth research question aims to analyse how effective the designed RA for deriving AAs was. For this, multi-case study research was applied in which three case studies to illustrate the application of the RA were undertaken. With the help of multi-case study research, the effectiveness of the proposed RA could be qualitatively assessed, taking the obstacles (available in Appendix C) from the previous study into account. The results of this research question are presented in the section "Evaluation of the reference architecture: a multi-case study approach".

Figure 5 shows how the research questions and their outputs are related. The first step was the SLR, which was done in the previous study. The SLR had the stakeholders and their concerns, features and obstacles as an output. In the second step, the actual design of an RA took place; this was started with the decision as to which views were needed based on the stakeholders and their concerns. If requirements were missing or not clear, a step back was made. The design of the RA had a set of views as output, which was evaluated with a multi-case study in the third step. The RA was evaluated with one case study with a project manager of an FMIS under development, one case study with a farmer that uses an FMIS daily and one with a developer of a commercially available FMIS. The case studies used the developed RA to make an AA of an existing FMIS. The information from the multi-case study was used again to improve the RA in the second step, and after the evaluation, the RA was finished.

Current reference architectures

The SLR from the previous study (Tummers et al. 2019) identified studies that presented an RA for FMISs and identified a few software architectures that can be used as a reference but are not named as such. The four prominent studies that presented RAs are presented in Table 2 and further discussed below.

Kruize et al. (2016) presented an RA that can be used to map, assess, design and implement farm software ecosystems that contribute to integrated FMISs. The farm software ecosystem in this study should allow for integration between ICT components on the farm.

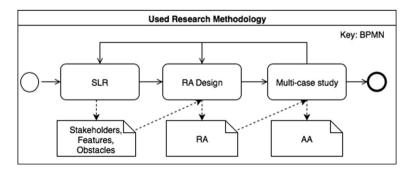


Fig. 5 The iterative method used in this study for deriving an architectural design



Study	Deviation from RA for FMISs	Views	
Kruize et al. (2016)	RA for Farm software ecosystems	Class diagram, deployment diagram	
López-Riquelme et al. (2017)	Not followed RA design approaches	Boxes-and-lines	
Kaloxylos et al. (2014)	Architecture for farm management system	Deployment diagram	
Nikkilä et al. (2010)	Not followed RA design approaches	Boxes-and-lines	

Table 2 The identified studies that present an RA for FMISs with their deviation from an RA for FMIS and their presented views

Based on requirement analysis, they presented a unified modelling language (UML) class diagram that described the relationships between the various components in the farm software ecosystem. These components were: the application component, the ICT component, the information system and the agri-food company. Furthermore, relationships between several actors of a farm software ecosystem were shown. These actors were: the software vendor, the agriculture service provider, the agri-food company, and the infrastructure provider. They presented the layout of the farm software ecosystem and showed how different components and actor roles were related to each other. From this diagram, it can be seen that both the end-user (the agri-food company) and sensor use the FMIS, both of which have their own application with their own user interface. Furthermore, it can be seen that the FMIS consists of multiple applications from multiple software vendors and that the FMIS contains a cloud node. The study from Kruize et al. (2016) focused on a structure that allowed for the integration of multiple FMISs, in contrast to this study that focuses on an RA for a single FMIS.

López-Riquelme et al. (2017) described a cloud-based software architecture for precision agriculture. They used FIWARE community (2020) as a cloud provider and used multiple components to develop an application that had the aim of reducing the amount of water needed for irrigation tasks. They presented the main hardware and software needed for this application. They furthermore presented a database structure and a graphical representation of the data flow. These diagrams are mainly focused on sensor data which is being sent via a GPRS node. The models in this study do, however, not follow formal modelling techniques [UML or comparable (Van Vliet 1993)] but give descriptions based on boxes and lines. In contrast to the goal of this study, the architecture given in López-Riquelme et al. (2017) was only focused on sending and receiving sensor data, which is only a part of the full FMIS.

Kaloxylos et al. (2014) presented a software architecture which can be considered as an RA for a cloud-based FMIS with a service-oriented architecture that could serve as a marketplace of services for farmers. They presented an ArchiMate (Open Group 2012) model of the system and also provided a proof of concept implementation. The FMIS was running on a cloud service and consisted of a monitoring service, data collector, data analyser, FMIS data storage service, notification Service, coordination module and an execution module. Furthermore, there was a local FMIS which had a logic controller, and there was a copy of the local FMIS. The cloud FMIS is connected with external services which include, monitoring reference services, solution reference services, resource management, scheduling and E-agriculturist service. Also, the equipment provider, pulling unit and implement, and the actuator are connected to the local FMIS. They performed an implementation of the architecture with field tests in a greenhouse. Kaloxylos et al. (2014) did



not go into depth of the FMIS architecture but presented the business logic for the case of a network connection failure, sensor problem and monitoring of service consumption. They offered much detail about the used server configuration, programming language, data schemas and databases used; this is, however of too much detail for the RA in this study.

Nikkilä et al. (2010) presented a software architecture for FMISs in precision agriculture. They started with the requirement analysis and, based on these requirements, they presented a software architecture that was structured as a web application. The FMIS should be able to create operational plans and store most of the data from the field operations as documentation. They presented the FMIS architecture from the viewpoint of the developer, starting with data storage and application logic, which are together considered as the FMIS. The application logic contains a communication, a data transformation, a class library, and a load balancing module. Data storage includes authentication, general FMIS and geographic information system (GIS) databases. The users, services, and authorities that use the FMIS via the data transfer (internet) are the human users, web services, ISO 11783 [International office for standardization (ISO) (2017] (allows communication with tractor-implement combinations) and other services. There was also a detailed view of the application logic presented, which contained the communication used, data transformation, class library and load balancing protocols and methods. This study, however, only showed two diagrams (based on boxes and lines) and the RA was mainly presented textually.

Reference architecture

In the following section the two selected sets of viewpoints that were used for documenting the RA are described. Subsequently, the FMIS RA using these two viewpoints is described. Finally, the method for deriving an AA from the proposed RA is discussed.

Selection of views

The main purpose of the FMIS is to assist in the management of the daily operations on the farm in the short term, but it should also determine the long-term vision for agricultural production (Cojocaru et al. 2014). To do so, the RA should be able to handle all features that have been derived from the previous study (Appendix B). The RA should be available for farmers in all different sectors of farming and be flexible to be customised for different sectors. For example, the needs for a dairy FMIS are of course different than the demands for an arable FMIS and thus will have different architecture decompositions.

For modelling the RA, the V&B approach from Clements et al. (2010) was adopted; this approach consists of a predefined set of viewpoints. The viewpoints can be applied for a broad set of applications and stakeholder concerns. The V&B approach describes more than 17 viewpoints, but as indicated in the V&B approach, only those viewpoints that are of interest are selected for modelling the architecture of a particular system. Two viewpoints were selected for modelling the FMIS RA that are of interest to almost any domain and also for the FMIS domain. These two considered viewpoints are the context viewpoint and the decomposition viewpoint. The context view of a system defines the scope of the system within the external environment. It depicts the entities that are outside the scope of the system but which have a direct relation with the system. Hence, it provides a particular systems' view in which the corresponding system is located in the overall scope from which it is part. The decomposition view describes the key elements of which the system



Fig. 6 The reference context diagram. This diagram shows the relation of the system to external entities. It ▶ shows the optional and required relations and entities. Only the interactions considered the most important (coming from the stakeholder concerns from Table 1) are shown

is composed. It shows how the system is decomposed into multiple modules and shows the sub-modules of these modules with a parent-child relationship.

Context view

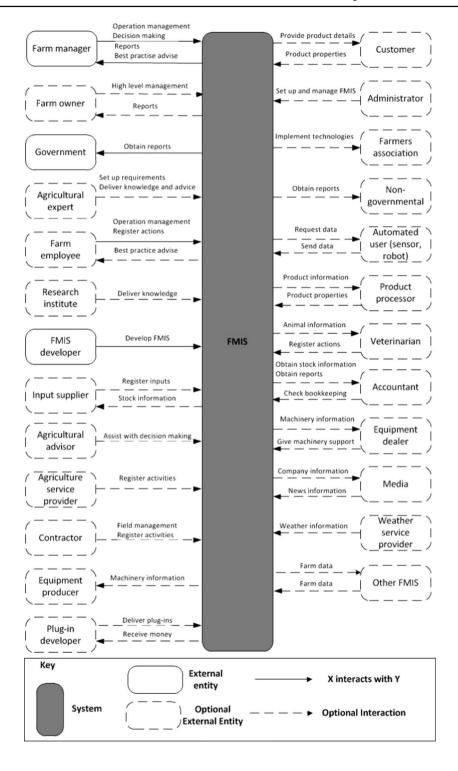
The context view of a system is represented using a so-called context diagram. This diagram represents the overall purpose of the system and its interfaces with an external environment (Kim et al. 2003). It shows the system boundaries, its environment and the entities it interacts with (Choubey 2011). It reveals what is in and what is out of the system and is often the first ingredient of architectural information for a reader. The context diagram for the FMIS is presented in Fig. 6. The external entities are based on the stakeholders from the "Background" section. Each farm owner has one or multiple farms that each have a farm manager. Furthermore, the automated user, the plug-in developer, and other FMISs were added as external entities. The automated user is described as a sensor (e.g., soil moisture sensor) or a robot (e.g., milk robot) that communicates with the sensor and provides data to the system. The plug-in developer is a developer from another company that delivers external software modules that can extend the system. The other FMIS entity presents an FMIS from another developer that interacts with the system.

The RA context diagram needs to map the variability in the external entities with the system. Not all external entities are always applicable to a certain FMIS, and certain external entities are not applicable to a specific sector of the agricultural domain. Tekinerdogan and Sözer (2012) presented methods for showing the variability in different views. Hereby, the mandatory elements are drawn with solid lines, while the optional elements are drawn using dotted lines. This approach was also applied in modelling the context view of the RA for FMIS, as shown in Fig. 6. The elements, (the modules as well as the relations) are based on the findings from the earlier SLR.

Decomposition view

As stated before, the decomposition view defines the decomposition of the system over different modules. Hereby, the only relation that is used is the decomposition relation which is usually shown by embedding a module in the overall system module. The view thus shows the decomposition of larger modules into smaller modules. This view often is the basis for project organization and system documentation (Van Vliet 1993). The decomposition view can help to extract user requirements because it gives a list of things you are supposed to ask when designing a software architecture. The decomposition view for the RA is presented in Fig. 7. This view shows all the possible modules for FMISs, and all modules are optional. The definition of the modules was a designer's issue based on the features from the previous SLR (available in Appendix B). The modules from the decomposition view should cover all features. The AA will consist of a selection of these modules divided over multiple subsystems.







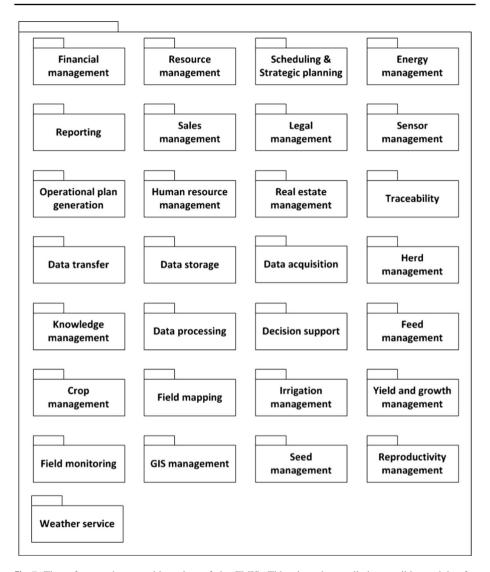


Fig. 7 The reference decomposition view of the FMIS. This view shows all the possible modules for FMISs, all modules are optional

Method for deriving an application architecture

As discussed at the beginning of the section and presented in Fig. 4, a specific view from the RA is reused to derive the corresponding view of the AA. The method for this derivation is presented in Fig. 8. First the requirements of the application are identified. Based on these requirements, features from the family feature model are selected. Based on the selected features, the required FMIS modules for the AA are selected from the FMIS RA. If the required module can be found in the RA, then this will be reused; if not, a new module will be added to the AA. In case the module is found in the RA, it is checked as



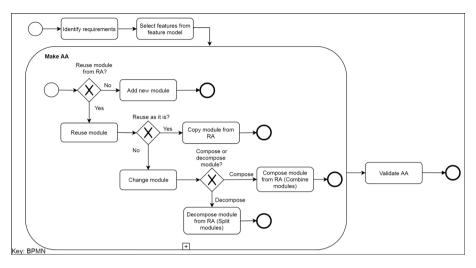


Fig. 8 Method for the derivation of the AA from the RA

to whether this module can be reused as-is, or if any adaptation is needed. A completely reusable module exists if the names of the modules are the same, or if the modules are interchangeable [e.g., financial management, vs. economic management (Yan-e 2012)]. If the module is not reusable as-is, and change is thus needed, the module can be composed or decomposed. In a composition, multiple modules from the RA are combined into one module in the AA. An example of a composition could be that the data transfer and data storage module are combined into a data management module (Murakami et al. 2013). In a decomposition, the module from the RA is split into numerous smaller modules in the AA. An example of decomposition could be that the livestock management module should be split up into a cow quality management, and a cow welfare support module (Berger and Hovay 2013). In the last process, the AA is validated to ensure a high-quality AA.

The application of two views for deriving the AA views was shown. In principle, the same process could be applied for other possible views (e.g., uses view, layered view etc.). For this, it is required to identify the required views based on the specific application requirements of the selected system (Clements et al. 2010). Once the required views are selected the approach discussed above will be the same; that is, first, the reference views will be developed after which the application views are derived.

Evaluation of the reference architecture: a multi-case study approach

The primary objective in this section was to evaluate the RA based on its practical use and effectiveness with retrospective and prospective case studies. This evaluation was done by developing an AA, with the method described in the "Reference architecture" section. For the evaluation of the RA, the case study approach was used following the guidelines from Runeson and Höst (2009). With the multi-case study, it was tried to validate and improve the proposed RA. The following five steps were followed for the case study: (1) case study design, (2) preparation for data collection, collecting evidence, (3) analysis of collected data, and (4) reporting. These five steps are further discussed below.



The primary purpose of this case study was to validate and improve the RA. Table 3 presents the design activities for the three selected case studies with as the primary goal the evaluation of the RA. The goal was the same for all case studies. The case studies aimed to answer the fourth research question of the study and identify the clearness and effectiveness of the RA. Data collection has been carried out by semi-structured interviews with end-user of FMISs (farmer), project managers, and FMIS developers. The data was analysed with a qualitative analysis of identified shortcomings of the RA and with a review of the determined differences of the FMISs with the RA.

Selected case studies and design

In this section, first, the selection of the case studies is described, and subsequently, the objective and planning of the case study are defined. An essential concept for the design of a case study is triangulation. Triangulation means taking multiple angles towards the subject of study (Runeson and Höst 2009). For this study, it was made sure there was data triangulation (having various cases) and observer triangulation (multiple companies/observers).

It was decided to use three different case studies. The first case study was conducted in collaboration with a farmer that uses an FMIS in practice. The second case study was performed in cooperation with a company that currently develops an FMIS, and the third case study was performed based on a commercially available FMIS.

For the first case study, a farmer was selected who uses a widely used FMIS for dairy farming in the Netherlands. The company that delivers this FMIS has solutions for the dairy, pig-husbandry, poultry and arable domain. The FMIS provider has more than 100 employees and delivers its product internationally. The selected dairy farmer is located in the north of the Netherlands and milks around 120 cows.

The second case is a prospective case performed in cooperation with an agricultural company active on the global market. This company is currently developing an FMIS for livestock farming, which makes this case a prospective case for an FMIS under development.

A third case study was performed with the head of technology from a commercial FMIS developer. The company 365FarmNet was chosen that was founded in 2013 and has over 80 employees. 365FarmNet presents a holistic software-based service (SaaS) that aims at the support of farmers in all aspects of farm management on their entire holding (365FarmNet 2020a). This FMIS is available as a web service, and it has an application for mobile phones and tablets. There is a basic free version, and add-ons from partner companies can be bought (365FarmNet 2020b). This FMIS was selected because it is considered as one of the most innovative systems currently available on the market.

Table 3 Case study design, adopted from (Runeson and Höst 2009)

Design activities	Description
Goal	Evaluate the RA for FMIS and derive an AA
Research Question	How effective is the designed RA for deriving AAs? (RQ4)
Data collection	Semi-structured interviews and architecture design
Data analysis	Analysis on identified short-comings/differences between AA and RA



Preparation for data collection

In this section, the procedures and protocols for the data collection are described. Data were collected with the help of semi-structured interviews. In the semi-structured interview, all questions were planned, but they were not necessarily asked in the same order as listed. The development of the interview decided in which order the questions were handled (Hove and Anda 2005). The list of the questions is available in Appendix D; for these questions, a distinguishing was made in questions for the farmer, the FMIS under development and 365FarmNet. The interviews were organised as follows and are based on Runeson and Höst (2009):

- 1. The objectives of the interview and the case study were presented. It was also explained how the data from the interview would be used.
- A set of introductory questions was asked about the background of the FMIS, which were relatively simple to answer.
- 3. The main interview questions were asked, which took up the largest part of the interview. First, the RA was evaluated, and the architectural design of their application (AA) was made with the help of the RA. For each presented view, the modelling technique and the logic behind the view were explained.

Using the methods presented in the "Reference architecture" section, the context diagram and decomposition view were derived. For the decomposition view, it was first decided in which sub-systems the application view should be divided. This division could, for example, be a division into an arable, dairy and data sub-system. Afterward, for each module from one of the case studies, it was checked if a module from the reference decomposition view could be re-used. This paper only shows the diagrams for the first case study for reasons regarding the length of the paper. Diagrams for the two other cases were also successfully made, but not presented in this paper.

Case study 1: farmer

The AA presented in this section is based on the specific use scenario of the farmer. The farmer mainly used the FMIS for registration purposes and thought the ability to print administration for himself and the government is most important. According to the perception of the farmer, communication with other organizations and integration with the feed computer in the barn could be improved, for example with extra modules. These modules are not presented in the AA derived since those modules are not used by the farmer and therefore the AA is limited. With the help of the overview of all modules in the RA the obstacle FMIS not complete (see Appendix C) could be solved in this case.

Context diagram

Figure 9 presents the context diagram for the case of the farmer. In comparison with Fig. 6, this diagram has fewer external entities. Multiple entities were not applicable to a dairy farmer, and others did not have a relationship with the FMIS in the case of the farmer.



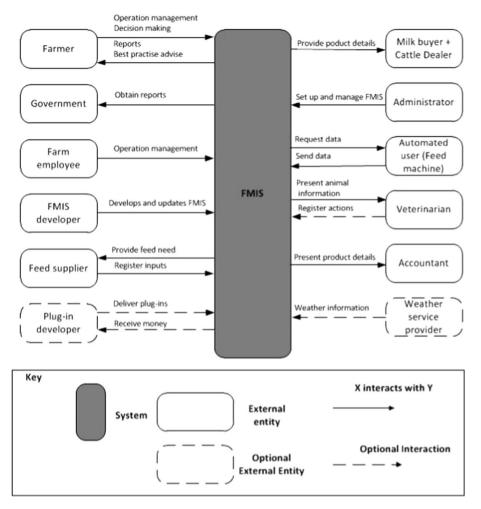


Fig. 9 The context diagram for the FMIS of the farmer. The diagram is based on the entities that interact with the system in the case of the farmer

Furthermore, the input supplier was re-named to the feed supplier, and the customer was decomposed into the milk buyer and cattle dealer.

Decomposition view

Figure 10 presents the decomposition view of the case of the farmer. The modules were selected from Fig. 7. Multiple elements could be copied one-to-one. The reporting module was decomposed in weekly and yearly reporting. The main functionality of the system is in the herd, reproductivity, and data management. Therefore, three sub-level decomposition views are presented in which these three modules are decomposed into sixteen smaller modules. In this view, there are no dependencies between the modules shown. However,



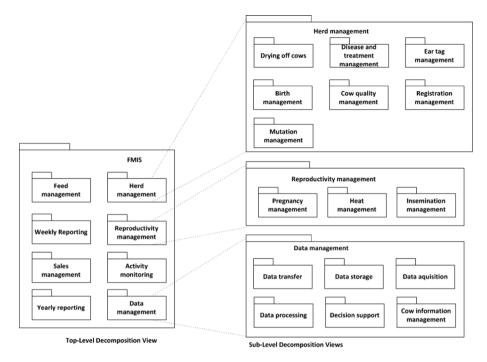


Fig. 10 The Decomposition view for the FMIS of the farmer. Only the components of the system that the farmer used are presented

for example, in practice, the heat management module can use the decision support module to predict fertility.

Case study 2: FMIS under development

With the help of the provided method for deriving an AA from the RA, a context diagram, and a decomposition view were made for the FMIS under development. The agricultural company mainly focuses on making an FMIS with basic features from which reports can be obtained. These reports can be examined and, based on this, the agricultural company can provide the farmer with advice. For the context diagram, the farm owner can have one or multiple farms, and each farm has its manager who is responsible for the daily operational management. In the context diagram, more interactions with the system were added, which were wanted by the agricultural company. The FMIS would consist of two main modules: herd management and feed management. Furthermore, some modules were made optional: sensor management, HR management, traceability management and financial management. These modules could be added later to the system or were optional for the farmer to choose from. A decomposition was needed for the herd, feed, and financial management to split them into smaller modules. A composition was needed for data management, which is composed of the modules data transfer, storage, acquisition and processing. For the development of the new FMIS, the obstacles listed in Appendix C should be taken into account.



Case study 3: 365FarmNet

The investigated FMIS was delivered as a platform and attached great importance to the input from add-on suppliers who provided extra modules to the platform. The main focus of the system was on documentation and supporting the user with decision making. Due to its platform structure, the system can overcome multiple obstacles listed in Appendix C, especially the system integration is better due to this structure. A context diagram and a decomposition view could be derived with the help of the RA and the methodology presented. For the context diagram, multiple entities were kept optional since these entities can be granted access to the system, but not necessarily have access to the system. Some names of entities needed to change to make the context diagram follow the company vocabulary. The decomposition view was split up into three different sub-level decomposition views: company, crop, and dairy. The company sub-level contained modules like human resource and stock management (decomposed from resource management), while the crop and dairy management sub-levels contained more production-related modules. Also, government linkage and add-on management modules were added to the decomposition sub-levels, since these were considered of key importance for the FMIS.

Overall results and findings

For all three case studies, the AA could be quickly derived from the RA. The answers to the semi-structured interviews and the overall discussion with the stakeholders showed that the method substantially simplified and speeded up the design of AAs. All stakeholders of the three cases indicated that they were not familiar with the feature diagram as well as the architecture view modelling approach. Hence, first this needed explanation but this did not take much effort in all of the cases. The stakeholders indicated the need for a more precise architecture modelling indeed and considered the approach both as practical and useful. The separation of the architecture views was considered useful since the current architecture diagrams that were used were usually described in informal, single diagrams, including the different concerns. The RA appeared to be understandable and practical for making the possible modules explicit. Compared to the required modules for the AA in the cases, the RA also appeared to be complete since no specific detailed models were needed to be added to the AA. The context view was considered very practical since it made the interaction with external entities explicit. While drawing the context diagram, an often made mistake was the need to define the interactions among the external entities themselves. The decomposition view was found simple and expressive and actually was somehow used in all the three cases, although not in a formal manner. Overall, the reuse approach, as provided by the method, was highly appreciated. Further, it was indicated that not only the RA by itself but also the overall process of deriving an AA from a feature diagram was helpful because it helped to discuss the design decisions in such an explicit manner. The stakeholders indicated that they would further use the method. An issue that was asked was whether other views existed, and this would be considered in future projects.



Discussion

This study presents a novel RA based on well-established architecture modelling approaches for FMISs using input from a previous literature review (Tummers et al. 2019). Thereby this study paves the way for similar studies on FMISs. From the results, several interesting observations could be identified.

With the help of the SLR from the previous study, four RAs were identified. Two of the four did not focus on the FMIS, but on the farm software ecosystem or farm management system, which have a far more comprehensive scope than FMIS (Kruize et al. 2016; Kaloxylos et al. 2014; Kassahun et al. 2016). Two others did not follow the ISO/ISEC/IEEE architecture standard (López-Riquelme et al. 2017; Nikkilä et al. 2010). Moreover, three out of the four RAs are not named as such in the identified literature; only Kruize et al. (2016) named the proposed architecture an RA. These observations might indicate that the field of (reference) architecture is a relatively new subject for researchers of FMISs.

Based on the observations in the previous paragraph, it was decided that there was a need to design a new RA (in alignment with Fig. 3). In the RA, only a set of two views could be proposed (given in the "Reference architecture" section) based on the literature input. If grey literature (e.g., software specification documents, website information, etc.) would have been utilised or expert interviews, there could have been another input for the requirements. This difference in the requirements could have led to a different selection of RA views or more views than the context diagram and decomposition view.

To derive the requirements for the context and decomposition view, the architecture design process was applied, which was proceeded by the requirement analysis. The stakeholders were identified and their requirements derived. The input for the reference context diagram was based on the stakeholders and their concerns coming from the literature. A disadvantage of a context diagram is that it only shows a limited amount of the interactions of the entities with the system. More interactions could be possible than the ones that are shown in the diagram. It was tried to mitigate this risk by only showing the essential interactions from the stakeholder concerns and by making multiple interactions optional.

The second view was the decomposition view. Although it was tried to keep this view as generic as possible, it is difficult to ensure that every necessary component is captured. The selection of components of the different views was a trade-off between the level of flexibility and level of re-use. It was tried to include as many components as possible by making most of them optional. It can however always be the case that some elements for a particular application are missing. It was tried to mitigate this risk by presenting the methods for the derivation of an AA in Fig. 8. These methods, however, allow for much change in the RA, which could make use of the RA less generic again.

The designed RA was used for deriving AAs with the help of a multi-case study approach. With the help of the derivation methods from Fig. 8, the AA could be mapped. Compositions and decompositions were needed, and some new modules were needed. This indicates that the RA, in combination with the provided methodology, was complete enough to map the case-studies. What was seen from the case studies is that FMISs in practise only focus on a (little) part of the RA, which is domain-dependent. For the multi-case study, there is always the threat of misinterpretation of different concepts. To mitigate this threat, the interpretation of the questions was verified with the interviewed persons. When the views from the case study were made, these were again



verified by the interviewed person to validate their correctness. From the retrospective and prospective case studies, it was identified that the RA and the methodology to derive an AA could be used to evaluate existing FMISs and guide the design of new FMISs.

It is shown that the methods defined by the software architecture design community can be used in the agri-food domain. It is also believed that the results of the study can be applied in a broader context than FMISs only. With the RA and the methods to derive AAs, future research in RA can be evaluated against well-established architecture design approaches. It is believed that in comparison with other RAs for FMISs, the RA is more generic and can be applied in more domains of the agricultural sector.

In the 1980s and the beginning of the 1990s, new software systems known as enterprise resource planning (ERP) systems revolutionised many large businesses. Companies such as SAP provided off the shelf solutions, which were tailored and implemented based on the company's requirements (Rashid et al. 2002). It is believed that such a revolution is also possible for FMISs. When off the shelf modules can be picked from an RA and can be combined into one FMIS based on the needs of the farmer, the concept of the FMIS can experience significant growth.

Conclusion

The RA presented in this study uses a different approach than other RAs for FMISs. The main objective was to present a new RA for FMIS based on the well-established software architecture design practices following the current software architecture standard (ISO/IEC/IEEE 42010 2011). Relating RAs for FMISs did not seem to follow this standard or proposed an RA for the agricultural domain with a scope broader than the FMIS alone. From the study, it becomes evident that the notion of architecture design and knowledge of modelling information systems in previous literature regarding FMISs can be considered weak.

Based on the inputs from the SLR in the previous study, the objective could be fulfilled, and an RA could be presented based on two architectural reference views: The context diagram and the decomposition view. With a multi-case study approach, FMIS AAs can be derived from the RA and thereby validate the RA and the methodology to derive AAs. The methods described in this study are generic and can be used for both developing a new RA and for enhancing the current RA with other views. The methods for the derivation of an RA are universal and can be used for the derivation of all possible views. To further improve the RA, it can be validated by performing more case studies with the methods presented in this study.

The presented RA is generic and can be used for FMISs in all sectors of the agricultural domain to overcome the FMIS obstacles and stakeholder concerns. The genericity is due to the flexibility of the system, which is based on the optionality of the components. With the help of the presented RA, researchers can identify the key research directions and practitioners can benefit from the results of this study by a thorough knowledge of the FMIS architecture. Different FMISs can be compared and new FMISs built.

In future work, the method will be applied in other industrial case studies. Also, a broader set of viewpoints and related agricultural domains will be considered in which the methods can be applied.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A: Included SLR studies

See Table 4.

Table 4 The 38 primary studies used as input for the SLR from Tummers et al. (2019)

Zheleva et al. (2017)	2017	Murakami et al. (2013)	2013	Paraforos et al. (2016)	2016
Hewage et al. (2017)	2017	Burlacu et al. (2013)	2013	Sørensen et al. (2010)	2010
Kjær (2008)	2008	Jiang and Zhang (2013)	2013	Husemann and Novković et al. (2014)	2015
Khaydarov et al. (2012)	2012	Yan-e (2012)	2012	Robbemond and Kruize (2011)	2011
Honda et al. (2014)	2014	Yu and Yongjun (2010)	2010	Bligaard (2014)	2014
López-Riquelme et al. (2017)	2017	Paraforos et al. (2017)	2017	Cojocaru et al. (2014)	2014
Kruize et al. (2016)	2016	Fountas et al. (2015a)	2015	Ampatzidis et al. (2016)	2016
Barmpounakis et al. (2015)	2015	Kaloxylos et al. (2014)	2014	Fountas et al. (2015b)	2015
Carli and Canavari (2013)	2013	Berger and Hovav (2013)	2013	Magne et al. (2010)	2010
Voulodimos et al. (2010)	2010	Kaloxylos et al. (2012)	2012	Allen and Wolfert (2011)	2011
Li et al. (2010)	2010	Sørensen et al. (2011)	2011	Tsiropoulos and Fountas (2015)	2015
Chen et al. (2016)	2016	Nikkilä et al. (2010)	2010	Tsiropoulos et al. (2013)	2013
Bojan et al. (2015)	2015	Tsiropoulos et al. (2017)	2017		



Appendix B: Features

See Table 5.

 Table 5
 Identified features in descending order. Adopted from Tummers et al. (2019)

Financial management	17	Expert knowledge	5	Transport management	2
Reporting	14	Livestock management	5	Calibration management	2
Data acquisition	13	Sales	5	Experience management	2
Operation plan generation	10	Harvest management	4	Marketing and sales	2
Crop management	10	Machinery tracking	4	Reproductivity management	2
Resource management	9	Pesticide management	4	Weighing management	2
Equipment management	9	Scheduling	4	Best practice	1
Field monitoring	9	Work management	4	Collect produce information	1
Data processing	9	Knowledge management	4	Communication	1
Fertilization management	9	Legal management	4	Condition management	1
Human resource management	7	Activity monitoring	4	Delivery management	1
Weather service	7	Customer management	4	Feed management	1
Data management	7	Alerting	4	Grazing management	1
Field management	7	Production monitoring	3	Herd management	1
Accounting	7	Seed management	3	Printing	1
Inventory management	7	Yield monitoring	3	Real estate management	1
Decision support	6	Parameter monitoring	3	Remote controlling	1
Operation management	6	Data sharing	2	Risk analysis	1
Yield estimation	6	Driver assistance	2	Society management	1
Field mapping	6	Energy management	2	Supply management	1
GIS management	6	Health management	2	Task file management	1
Irrigation management	6	Information search	2	Task supervision	1
Sensor management	6	Model production parameters	2	Vision planning	1
Traceability	6	Performance management	2	B2B Collaboration	1
Data transfer	5	Scenario simulation	2	Company information	1
Data storage	5	Strategic planning	2	Environmental monitoring	1
Disease management	5	Technology management	2	Planting management	1



Appendix C: Obstacles

See Table 6.

Table 6 Identified obstacles in descending order. Adopted from Tummers et al. (2019)

Standardised data formats	9	No standardised solution	2	Lack of interoperability standards	1
System Integration	9	Poor user interface	2	Lack of need for farmer	1
Adaption rate of FMIS	7	Scalability	2	Large input needed	1
Cost of FMIS	7	Time consumption	2	Missing modules	1
FMIS not complete	7	Access to system	1	Model integration	1
Not easy to understand	7	Data accuracy	1	No expert trust	1
Connection to internet	3	Data confidentiality	1	No modularity	1
Data size	5	Data availability	1	No farmer specific design	1
Insufficient farmer skills	3	Data correctness	1	Not applicable in other context	1
Language and regional	3	Data exchange	1	Not service oriented	1
Security	3	Data intensity	1	Performance assurance	1
Availability	2	Data is scattered	1	Privacy	1
Complexity of farms	2	Data management	1	Proprietary system	1
Data complexity	2	Data processing	1	Reliability	1
Data diversity	2	Data timeliness	1	Slow information updating	1
Data integration	2	Early stage of development	1	Too much information	1
Data integrity	2	Knowledge integration	1	Too specialised	1
Different interfaces	2	Lack of information standards	1		

Appendix D: Semi-structured interview questions

Introductory questions.

Farmer

- a. What is your experience with FMISs?
- b. For which purposes do you use your FMIS?
- c. What do you find most important about your FMIS?
- d. Which feature of the FMIS do you use the most?
- e. How easy do you find your FMIS to use?
- f. Which problems do you encounter with the use of the FMIS?
- g. What is your FMIS missing?

FMIS under development

- 1. How would you describe an FMIS?
- 2. What is the main reason for developing an FMIS?
- 3. What do you consider most important for an FMIS?
- 4. Where are most of your customers situated?



- 5. In which agricultural sectors are you currently active and to which sectors do you want to expand?
- 6. What do you think the most used feature of the FMIS will be?

365FarmNet

- 1. How would you describe an FMIS?
- 2. What do you consider most important for an FMIS?
- 3. Where are most of your customers situated?
- 4. In which agricultural sectors are you currently active and to which sectors do you want to expand?
- 5. How important do you think the add-ons are for your system?
- 6. What is the most used feature of the FMIS?

Main questions.

Farmer, FMIS under development & 365FarmNet

- 1. Do you miss any relationship with external entities in the context diagram?
- 2. Do you think some relationships in the context diagram are unnecessary?
- 3. Make the context diagram based on methods in the "Reference architecture" section.
- 4. Do you think some modules of the decomposition view are unnecessary?
- Make the decomposition view based on methods in the "Current reference architectures" section
- 6. Are you missing an architectural view?
- 7. How do you think this RA can be improved?

References

- 365FarmNet (2020a). About us. Retrieved May 14, 2020 from https://www.365farmnet.com/en/company/about-us/.
- 365FarmNet (2020b). Cooperation and integration. Retrieved May 14, 2020 from https://www.365farmnet.com/en/365partner/.
- Allen, J., & Wolfert, J. (2011). Farming for the future: Towards better information-based decision-making and communication-Phase I: Australasian stocktake of farm management tools used by farmers and rural professionals. New Zealand Centre of Excellence in Farm Business Management, Palm. Technical Report, AgFirst Consultancy/Wageningen University and Research Centre.
- Ampatzidis, Y., Tan, L., Haley, R., & Whiting, M. D. (2016). Cloud-based harvest management information system for hand-harvested specialty crops. Computers and Electronics in Agriculture, 122, 161–167.
- Barmpounakis, S., Kaloxylos, A., Groumas, A., Katsikas, L., Sarris, V., Dimtsa, K., et al. (2015). Management and control applications in agriculture domain via a future internet business-to-business platform. *Information Processing in Agriculture*, 2(1), 51–63.
- Bass, L., Clements, P., & Kazman, R. (2003). Software architecture in practice. Boston, USA: Addison Wesley Professional.
- Berger, R., & Hovav, A. (2013). Using a dairy management information system to facilitate precision agriculture: The case of the AfiMilk® system. *Information Systems Management*, 30(1), 21–34.
- Bligaard, J. (2014). Mark online, a Full Scale GIS-based Danish farm management information system. *International Journal on Food System Dynamics*, 5(4), 190–195.



- Bojan, V.-C., Raducu, I.-G., Pop, F., Mocanu, M., & Cristea, V. (2015). Cloud-based service for time series analysis and visualisation in farm management system. In 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 425–432. IEEE
- Burlacu, G., Cojocaru, L.-E., Danila, C., Popescu, D., & Stanescu, A. M. (2013). A digital business ecosystem integrated approach for farm management information system. In 2013 2nd International Conference on Systems and Computer Science (ICSCS), pp. 80–85. IEEE.
- Capterra (2020). Farm management software. Retrieved May 14, 2020, from https://www.capterra.com/farm-management-software/.
- Carli, G., & Canavari, M. (2013). Introducing direct costing and activity based costing in a farm management system: A conceptual model. *Procedia Technology*, 8, 397–405.
- Chen, P.-J., Du, Y.-C., Cheng, K.-A., & Po, C. Y. (2016). Development of a management system with RFID and QR code for matching and breeding in Taiwan pig farm. In 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1–5. IEEE.
- Choubey, M. K. (2011). *IT infrastructure and management (for the GBTU and MMTU)*. Delhi, India: Pearson Education India.
- Clements, P., Garlan, D., Little, R., Nord, R., Stafford, J., Bachmann, F., et al. (2010). *Documenting software architectures: Views and beyond* (2nd ed.). Boston, USA: Addison-Wesley Professional.
- Cojocaru, L.-E., Burlacu, G., Popescu, D., & Stanescu, A. M. (2014). Farm Management Information System as ontological level in a digital business ecosystem. Service orientation in Holonic and multi-agent manufacturing and robotics (pp. 295–309). Cham, Switzerland: Springer.
- FIWARE community (2020). FIWARE: The open source platform for our smart digital future. Retrieved May 14, 2020 from https://www.fiware.org/.
- Fountas, S., Carli, G., Sørensen, C., Tsiropoulos, Z., Cavalaris, C., Vatsanidou, A., et al. (2015a). Farm management information systems: Current situation and future perspectives. *Computers and Electronics in Agriculture*, 115, 40–50.
- Fountas, S., Sorensen, C. G., Tsiropoulos, Z., Cavalaris, C., Liakos, V., & Gemtos, T. (2015b). Farm machinery management information system. Computers and Electronics in Agriculture, 110, 131–138.
- Hewage, P., Anderson, M., & Fang, H. (2017). An Agile Farm Management Information System Framework for Precision Agriculture. In *Proceedings of the 9th international conference on information manage*ment and engineering, pp. 75–80. ACM, New York, USA: Association for Computing Machinery.
- Honda, K., Ines, A. V. M., Yui, A., Witayangkurn, A., Chinnachodteeranun, R., & Teeravech, K. (2014). Agriculture information service built on geospatial data infrastructure and crop modelling. In *Proceedings of the 2014 international workshop on web intelligence and smart sensing*, pp. 1–9. ACM, New York, USA: Association for Computing Machinery.
- Hove, S. E., & Anda, B. (2005). Experiences from conducting semi-structured interviews in empirical software engineering research. In *Proceedings international software metrics symposium*, pp. 10–23. Picataway, USA: IEEE.
- Husemann, C., & Novković, N. (2014). Farm management information systems: A case study on a German multifunctional farm. *Economics of Agriculture*, 61(2), 441–453.
- ISO/IEC/IEEE 42010 (2011). Systems and software engineering–architecture description. Technical Report, ISO/IEC/IEEE 42010.
- International office for standardization (ISO) (2017). ISO 11783-1:2017 Tractors and machinery for agriculture and forestry—serial control and communications data network—Part 1: General standard for mobile data communication. Retrieved May 14, 2020, from https://www.iso.org/standard/57556.html.
- Jiang, R., & Zhang, Y. (2013). Research of agricultural information service platform based on internet of things. In 2013 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science, pp. 176–180. IEEE.
- Kaloxylos, A., Eigenmann, R., Teye, F., Politopoulou, Z., Wolfert, S., Shrank, C., et al. (2012). Farm management systems and the Future Internet era. Computers and Electronics in Agriculture, 89, 130–144.
- Kaloxylos, A., Groumas, A., Sarris, V., Katsikas, L., Magdalinos, P., Antoniou, E., et al. (2014). A cloud-based farm management system: Architecture and implementation. Computers and Electronics in Agriculture, 100, 168–179.
- Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., Peterson, A. S. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report, Carnegie-Mellon University, Pittsburgh, PA, USA: Software Engineering Inst.
- Kassahun, A. (2017). Aligning business processes and IT of multiple collaborating organisations. PhD Thesis, Wageningen University, The Netherlands.



- Kassahun, A., Hartog, R. J. M., & Tekinerdogan, B. (2016). Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture. *Computers and Electronics in Agriculture*, 123, 275–291.
- Khaydarov, Z., Laine, T. H., Gaiani, S., Choi, J., & Lee, C. (2012). Context-aware agriculture organizer. In Proceedings of the 6th international conference on ubiquitous information management and communication, ICUIMC'12, p. 69. New York, USA: Association for Computing Machinery.
- Kim, C. H., Weston, R. H., Hodgson, A., & Lee, K. H. (2003). The complementary use of IDEF and UML modelling approaches. *Computers in Industry*, 50(1), 35–56.
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and Software Technology*, 51(1), 7–15.
- Kjær, K. E. (2008). Designing middleware for context awareness in agriculture. In *Proceedings of the 5th middleware doctoral symposium*, pp. 19–24. ACM, New York, USA: Association for Computing Machinery.
- Köksal, O., & Tekinerdogan, B. (2019). Architecture design approach for IoT-based farm management information systems. *Precision Agriculture*, 20(5), 926–958.
- Kruize, J. W., Wolfert, J., Scholten, H., Verdouw, C. N., Kassahun, A., & Beulens, A. J. (2016). A reference architecture for Farm Software Ecosystems. *Computers and Electronics in Agriculture*, 125, 12–28.
- Li, M., Qian, J. P., Yang, X. T., Sun, C. H., & Ji, Z. T. (2010). A PDA-based record-keeping and decisionsupport system for traceability in cucumber production. *Computers and Electronics in Agriculture*, 70(1), 69–77.
- López-Riquelme, J. A., Pavón-Pulido, N., Navarro-Hellín, H., Soto-Valles, F., & Torres-Sánchez, R. (2017).
 A software architecture based on FIWARE cloud for Precision Agriculture. Agricultural Water Management, 183, 123–135.
- Magne, M. A., Cerf, M., & Ingrand, S. (2010). A conceptual model of farmers' informational activity: A tool for improved support of livestock farming management. *Animal*, 4(6), 842–852.
- Muller, G. (2012). A reference architecture primer, White paper. Eindhoven, The Netherlands: Eindhoven University of Technology.
- Murakami, Y., Utomo, S. K. T., Hosono, K., Umezawa, T., & Osawa, N. (2013). IFarm: Development of cloudbased system of cultivation management for precision agriculture. In 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE), pp. 233–234. IEEE.
- Nikkilä, R., Seilonen, I., & Koskinen, K. (2010). Software architecture for farm management information systems in precision agriculture. *Computers and Electronics in Agriculture*, 70(2), 328–336.
- Open Group. (2012). ArchiMate 2.0 specification: Open Group Standard. 's Hertogenbosch, The Netherlands: Van Haren Publishing.
- Paraforos, D. S., Vassiliadis, V., Kortenbruck, D., Stamkopoulos, K., Ziogas, V., Sapounas, et al. (2016). A Farm Management Information System using Future Internet technologies. *International Federation of Automatic Control (IFAC)-PapersOnLine*, 49(16), 324–329.
- Paraforos, D. S., Vassiliadis, V., Kortenbruck, D., Stamkopoulos, K., Ziogas, V., Sapounas, A. A., et al. (2017). Multi-level automation of farm management information systems. *Computers and Electronics in Agriculture*, 142, 504–514.
- Project Management Institute. (2013). A guide to the project management body of knowledge (PMBOK ® guide). Newtown Square, USA: Project Management Institute.
- Rashid, M. A., Hossain, L., & Patrick, J. D. (2002). The evolution of ERP systems: A historical perspective. Enterprise resource planning: Solutions and management (pp. 35–50). Hershey, USA: IGI Global.
- Robbemond, R., & Kruize, J. W. (2011). Data standards used for data-exchange of FMIS (44). Technical Report, Wageningen, The Netherlands: Wageningen Economic Research.
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in soft-ware engineering. *Empirical Software Engineering*, 14(2), 131–164. https://doi.org/10.1007/s1066 4-008-9102-8.
- Sørensen, C. G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S. M., et al. (2010). Conceptual model of a future farm management information system. *Computers and Electronics in Agriculture*, 72(1), 37–47.
- Sørensen, C. G., Pesonen, L., Bochtis, D. D., Vougioukas, S. G., & Suomi, P. (2011). Functional requirements for a future farm management information system. *Computers and Electronics in Agriculture*, 76(2), 266–276.
- Stafford, J. V. (2000). Implementing precision agriculture in the 21st century. Journal of Agricultural Engineering Research, 76(3), 267–275.



- Tekinerdogan, B., & Köksal, O. (2018). Pattern based integration of internet of things systems. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) (pp. 19–33). Cham, Switzerland: Springer.
- Tekinerdogan, B., & Sözer, H. (2012). Variability viewpoint for introducing variability in software architecture viewpoints. In Proceedings of the working international conference on software architecture (WICSA) & European conference on software architecture (ESCA) 2012 Companion Volume, pp. 163–166. ACM, New York, USA: Association for Computing Machinery.
- Tekinerdogan, B., & Uzun, B. (2019). Design of variable big data architectures for E-Government Domain. Software engineering for variability intensive systems: Foundations and applications (pp. 251–274). Cleveland, USA: Chemical Rubber Company Press.
- Tsiropoulos, Z., & Fountas, S. (2015). Farm management information system for fruit orchards. In Stafford, J. V. (Ed.) *Precision Agriculture'15, Proceedings of the 10th European Conference on Precision Agriculture*, pp. 44–55. Wageningen, The Netherlands: Wageningen Academic Publishers.
- Tsiropoulos, Z., Fountas, S., Gemtos, T., Gravalos, I., & Paraforos, D. (2013). Management information system for spatial analysis of tractor-implement draft forces. In Stafford, J. V. (Ed.) *Precision Agriculture'13, Proceedings of the 9th European Conference on Precision Agriculture*, pp. 349–356. Wageningen, The Netherlands: Wageningen Academic Publishers.
- Tsiropoulos, Z., Carli, G., Pignatti, E., & Fountas, S. (2017). Future perspectives of Farm Management Information Systems. *Precision agriculture: Technology and economic perspectives* (pp. 181–200). Cham, Switzerland: Springer.
- Tummers, J., Kassahun, A., & Tekinerdogan, B. (2019). Obstacles and features of Farm Management Information Systems: A systematic literature review. Computers and Electronics in Agriculture, 157, 189–204.
- US Dept. of Defence/Office of the DoD CIO (2010). Reference architecture description. Technical Report June, US Dept. of Defence/Office of the DoD CIO. Retrieved May 14, 2020, from https://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf.
- Van Vliet, H. (1993). Software engineering: Principles and practice (Vol. 3). New York, USA: Wiley.
- Voulodimos, A. S., Patrikakis, C. Z., Sideridis, A. B., Ntafis, V. A., & Xylouri, E. M. (2010). A complete farm management system based on animal identification using RFID technology. *Computers and Elec*tronics in Agriculture, 70(2), 380–388.
- Waston, H., Caroll, A., & Mann, R. (1991). *Information Systems for Management: A book of readings*. Homewood, USA: Richard D Irwin.
- Yan-e, D. (2012). Research about based-SOA agriculture management information system. In 2012 International Conference on Information and Automation (ICIA), pp. 78–82. IEEE.
- Yu, L., & Yongjun, L. (2010). A research and practice for sugarcane area's farm management information service platform. In 2010 International Conference on Computer Application and System Modeling (ICCASM), (vol. 9, pp. V9-218). IEEE.
- Zheleva, M., Bogdanov, P., Zois, D.-S., Xiong, W., Chandra, R., & Kimball, M. (2017). Smallholder agriculture in the information age: Limits and opportunities. In *LIMITS 2017—Proceedings of the 2017 workshop on computing within limits*, pp. 59–70, New York, USA: Association for Computing Machinery.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

